

## CLAIMS

What is claimed is:

- 1 1. A method of preventing an attack on a network, wherein the attack comprises  
2 injecting a spurious transmission control protocol (TCP) segment into a TCP connection  
3 between a sender and a receiver, the method comprising the computer-implemented steps of:  
4 receiving a duplicate TCP ACK message;  
5 incrementing a false duplicate ACK counter when a TCP re-transmission buffer  
6 maintained by the receiver is empty;  
7 when the false duplicate ACK counter is equal to a specified strike factor, sending a  
8 corrective ACK message that provides a correct sequence value and ACK  
9 value.
  
- 1 2. A method as recited in Claim 1, wherein the specified strike factor is a value in a  
2 range of 1 to 10.
  
- 1 3. A method as recited in Claim 1, wherein the steps are performed by an endpoint node  
2 acting as the receiver of data in the TCP connection.
  
- 1 4. A method as recited in Claim 1, wherein the steps are performed by a TCP  
2 application of an operating system of a network infrastructure element.
  
- 1 5. A method as recited in Claim 1, wherein the steps are performed by a TCP process,  
2 stack, adapter or agent hosted by or associated with an operating system of a personal  
3 computer, workstation or other network end station.
  
- 1 6. A method as recited in Claim 1, further comprising the steps of:  
2 receiving the corrective ACK message;  
3 determining whether the correct sequence value is less than another sequence value of  
4 a segment in a re-assembly buffer;

5 discarding the segment from the re-assembly buffer when the correct sequence value  
6 is less than the other sequence value of a segment in the re-assembly buffer.

1 7. A method as recited in Claim 6, wherein the discarding step comprises discarding all  
2 segments in the re-assembly buffer.

1 8. A method of preventing an attack on a network, wherein the attack comprises  
2 injecting a spurious transmission control protocol (TCP) segment into a TCP connection  
3 between a sender and a receiver, the method comprising the computer-implemented steps of:  
4 receiving a particular TCP segment;  
5 determining a sequence value gap as between the particular TCP segment and a prior  
6 TCP segment previously placed in a re-assembly buffer maintained by the  
7 receiver;  
8 determining whether the sequence value gap is too large according to a specified  
9 heuristic;  
10 if the sequence value gap is too large, then performing the steps of:  
11 creating and sending a dummy segment carrying a particular sequence value  
12 that is just prior to a last properly acknowledged sequence value;  
13 receiving an acknowledgment of the dummy segment;  
14 determining whether a second sequence value carried in the acknowledgment  
15 is less than a third sequence value of the first TCP segment; and  
16 discarding the particular TCP segment from the re-assembly buffer when the  
17 second sequence value carried in the acknowledgment is less than the  
18 third sequence value of the particular TCP segment.

1 9. A method as recited in Claim 8, wherein the specified heuristic holds when a re-  
2 assembly gap value is greater than one-half of a then-current window size.

1 10. A method as recited in Claim 8, wherein the specified heuristic holds when a re-  
2 assembly gap value is greater than the lesser of (a) one-half of a then-current window size or  
3 (b) a multiple of a maximum allowed segment size.

1       11.     A method as recited in Claim 8, wherein the specified heuristic holds when a query  
2     threshold is greater than a function of then-current bandwidth and then-current delay.

1       12.     A method as recited in Claim 8, wherein the specified heuristic holds when a query  
2     threshold is greater than a function of a round-trip time (RTT) estimate, a then-current  
3     available bandwidth value, and window size.

1       13.     A method as recited in Claim 8, wherein the specified heuristic holds when a query  
2     threshold value is less than or equal to a re-assembly gap value and the re-assembly gap  
3     value is less than or equal to a receive window value.

1       14.     A method as recited in Claim 13, wherein the re-assembly gap value comprises a  
2     sequence value of a first segment in the re-assembly buffer less a next expected segment  
3     value.

4       15.     An apparatus for preventing an attack on a network, wherein the attack comprises  
5     sending a spurious transmission control protocol (TCP) segment with a spurious or unwanted  
6     DATA segment, comprising means for performing any of the steps of Claims 1, 2, 3, 4, 5, 6,  
7     7, 8, 9, 10, 11, 12, 13, or 14.

1       16.     An apparatus for preventing an attack on a network, wherein the attack comprises  
2     sending a spurious transmission control protocol (TCP) segment with spurious or  
3     unwanted DATA segment, comprising:  
4              a processor;  
5              one or more stored sequences of instructions that are accessible to the processor and  
6              which, when executed by the processor, cause the processor to carry out the  
7              steps of any of Claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, or 14.

1    17. A computer-readable medium carrying one or more sequences of instructions for  
2 preventing an attack on a network, wherein the attack comprises sending a spurious  
3 transmission control protocol (TCP) segment with unwanted or spurious DATA , wherein the  
4 execution of the one or more sequences of instructions by one or more processors causes the  
5 one or more processors to perform the steps of any of Claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,  
6 12, 13, or 14.